Amendments to the Claims:

This listing of claims will replace all prior versions and listing of claims in the application:

Listing of Claims

1.   (Currently amended) A client server system using distributed objects, comprising:

a client connected to a communication network for performing an access request to an object;

an application server for performing an application by an actual object according to the access request by said client; and

an object pool server connected to said client through said communication network and connected to said application server for pooling a proxy object corresponding to said actual object and for holding actual object management information that is [[the]] information on said actual object, wherein said application server notifies said object pool server of an event according to a change in status of said application, and said object pool server automatically updates said actual object management information according to the notification of said event from said application server.

2.   (Original) The client server system as set forth in Claim 1, wherein the event notified from said application server is formed according to at least one of the result of a process of starting a project and the result of a process of stopping the project.

3.   (Original) An object pool using distributed objects, comprising:

a client request analyzing unit for analyzing an access request to an object;

an object information storage unit for storing object information at the termination process of said object pool;

an object creating unit for creating an object at the starting process of said object pool according to said object information stored by said object information storage unit; and

an object managing unit for pooling the object created by said object creating unit before accessing said object from said client.

4. (Original) The object pool as set forth in Claim 3, wherein the object information stored by said object information storage unit is constructed so that it can be at least recognized to be the last accessed object, and said object creating unit starts creation from said last accessed object.

5. (Original) A client server system using distributed objects, comprising:

an object pool connected through a communication network to a client which sends an access request to an object, and for pooling objects and managing object information; and

an application execution environment connected to said object pool for executing an application according to the access request of said client, and notifying said object pool of an event according to a status change of said application, wherein said object pool updates said object information according to the event notification from said application execution environment.

6. (Original) The client server system as set forth in Claim 5, wherein an object pool server having the function of said object pool and an application server in said application execution environment are connected to each other through a network or the like, said object pool server pooling objects as proxy objects.

7. (Original) The client server system as set forth in Claim 5, wherein said object pool and said application execution environment are formed on the same server.

8. (Original) An object pooling method for pooling objects in advance on a specified server to execute an application in a distributed system, said object pooling method comprising the steps of:

storing object information when a process of said server is terminated; and

creating objects according to said object information when a process of said server is started; and

pooling the created objects.

9. (Original) The object pooling method of Claim 8, wherein said object information is stored with a predetermined priority, and said objects are created in descending order with respect to said priority.

10. (Original) An object pooling method on an object pool which is connected through a communication network to a client which sends an access request to an object, and is connected to an application execution environment in which an actual object is executed, comprising the steps of:

pooling a proxy object corresponding to an actual object in said application execution environment;

recognizing a change in status of the actual object in said application execution environment; and

updating actual object management information according to said recognized status change.

11. (Original) A storage medium in which a program to be executed by a computer is stored so that it can be read by an input unit of said computer, wherein said program makes said computer execute:

a monitoring process for monitoring execution status of a project which executes an application utilizing distributed objects;

an event creating process for creating an event according to execution status of the project monitored by said monitoring process; and

an event issuing process for issuing the event created by said event creating
process to an object pool server connected through a network or the like.

12.   (Original) A storage medium in which a program to be executed by a computer is
stored so that it can be read by an input unit of said computer, said computer being
connected through a network or the like to an application server for executing an
application utilizing distributed objects, and to a client which requests the execution of
said application, wherein said program makes said computer execute:
        an object pooling process for pooling the proxy objects corresponding to actual
objects to be executed by said application server; and
        an updating process for receiving an event issued from said application server,
and updating management information on said actual object according to said event.

13.   (Original) The storage medium as set forth in Claim 12, wherein said computer is
made to further execute:
        an execution status managing process for keeping track of execution statuses of
said objects, and managing them for each object;
        a request analyzing process for analyzing a request including an object creation
request and/or an object deletion request from said client; and
        a process for executing, upon receipt of a result of a request analysis by said
request analyzing process, creation of an object and/or the deletion of an object according
to a management result by said execution status managing process.

14.   (Original) A program sending apparatus, comprising:
        a storage unit for storing a software product which makes a computer execute an
event forming program for forming an event according to a change in status of an
application utilizing distributed objects, and an object pooling program for pooling
objects according to the event formed by said event forming process; and
        a sending unit for reading out said program from said storage unit, and sending
said software product.

15.   (Original) A program sending apparatus, comprising:

a storage unit for storing a program which makes a computer execute an object pooling process for pooling, on a server, objects associated with execution of an application utilizing distributed objects, an information storing process for storing object information in said server, and a creation sequence determining process for determining a sequence of objects to be created according to said object information stored by said information storing process; and

a sending unit for reading out said program from said storage unit, and sending said program.